SPECIFICATION AMENDMENTS

1. On page 1, lines 22-27, please amend as follows:

"The present U.S. Utility Patent Application is also a continuation-in-part of U.S. Utility Patent Application Serial No. 10/723,574, entitled "LDPC (Low Density Parity Check) coded modulation hybrid decoding," (Attorney Docket No. BP3134), filed November 26, 2003 (11/26/2003), ~~pending~~ now U.S. Patent 7,185,270 B2, issued on February 27, 2007 (02/27/2007), which is hereby incorporated herein by reference in its entirety and made part of the present U.S. Utility Patent Application for all purposes."

2. On page 33, lines 8-15, please amend as follows:

"An irregular LDPC code may also be described using a bipartite graph. However, the degree of each set of nodes within an irregular LDPC code may be chosen according to some distribution. Therefore, for two different variable nodes, v_{i_1} and v_{i_2}, of an irregular LDPC code, $|E_v(i_1)|$ may not equal to $|E_v(i_2)|$. This relationship may also hold true for two check nodes. The concept of irregular LDPC codes was originally introduced within M. Luby, M. Mitzenmacher, M. A. Shokrollahi, D. A. Spielman, and V. Stemann, "Practical Loss-Resilient Codes," *Proc. 29th Symp. on Theory of Computing*, 1997, pp. 150-159 ~~M. Lugy, M. Mitzenmacher, A. Shokrollahi, D. Spielman and V. Stemann, "Practical loss-resilient codes," *IEEE Trans. Inform. Theory*, Vol. 47, pp. 569-584, Feb. 2001~~."

3. On page 51, line 22 until page 55, 28, please amend as follows:

"When performing the update of the edge messages within the check node update functional block, the check node update functional block updates the check ~~bit~~ edge messages using the updated bit edge messages passed by the last iteration. During a first decoding iteration, this may include using the initialized values of the bit edge message. However, during the iterative decoding processing, the check node update functional block passed the updated edge messages to the symbol sequence estimate and symbol node update functional block.

Again, it is noted that the symbol sequence estimate and symbol node update functional block uses the initial conditions of the LLR bit edge message during its first iteration of the iterative decoding processing. It also uses the initially received symbol metric value during subsequent iterations of the iterative decoding processing. The symbol sequence estimate and symbol node update functional block initially performs computation of the possible soft symbol estimates. Then, the symbol sequence estimate and symbol node update functional block uses this information to assist in the updating of the edge messages. More specifically, the symbol sequence estimate and symbol node update functional block updates the bit edge messages using the computed symbol metric (from the m-bit symbol metric computer) combined with the check ~~bit~~ edge messages ~~message~~ passed by the last iteration from the check node update functional block. From one perspective, this shows the hybrid decoding functionality such that a combined use of bit level information and symbol level information are both used in a manner that (as is also described below) that provides a significant reduction in complexity and ease of implementation while providing performance that may be as good as the symbol decoding approach in some embodiments. In general, the performance of the hybrid decoding approach is as good as or worse than the symbol decoding approach; however, the hybrid decoding approach may be implemented significantly easier than the symbol decoding approach (e.g., with significantly reduced processing, memory, and memory management resources).

The iterative decoding processing continues between the symbol sequence estimate and symbol node update functional block and the check node update functional block such that the edge messages are continually, successively and alternatively updated in an effort to converge on a final value of the bit edge messages (either after performing a predetermined number of iterations or after a sufficient degree of accuracy is achieved ~~and the bit edge messages have converged on a final value, that meets the sufficient degree of accuracy~~). The updating is successive and alternative from the perspective that the symbol sequence estimate and symbol node update functional block performs an updating, and then the check node update

functional block performs an updating, and then the iterative decoding processing continues."

4. On page 56, line 13 until page 57, line 2, please amend as follows:

"FIG. 27 is a diagram illustrating another embodiment of hybrid decoding functionality (having a reduced complexity when compared to symbol decoding) of LDPC coded modulation signals according to the invention. In this alternative implementation, these iterative decoding processing steps are repeated until the syndromes of the LDPC code are all equal to zero (within a certain degree of precision). As mentioned above, soft symbol estimate is generated within the symbol sequence estimate and symbol node update functional block. This soft output information may be provided to a hard limiter where hard decisions may be made, and that hard information may be provided to a syndrome calculator to determined whether the syndromes of the LDPC code are all equal to zero (within a certain degree of precision). When they are not, a syndrome check failed signal may be provided to the iterative decoding processing functional block (and when it is determined that this decoding iteration is not the last decoding iteration), and the iterative decoding processing continues again by appropriately updating and passing the bit edge messages and the check edge messages between the check node update functional block and the symbol sequence estimate and symbol node update functional block. After all of these iterative decoding processing steps have been performed, then the best estimates of the bits are output based on the soft symbol estimates. It is also noted that some additional decisions and/or operations may be implemented when the situation arises where the syndromes of the LDPC code are not converging substantially to zero (within a certain degree of precision) and yet the last decoding iteration has in fact been performed."

5. On page 57, line 20 to line 27, please amend as follows:

"The method then continues by performing the check node updating. This check node updating involves updating the check ~~bit~~ edge messages using the bit edge

messages passed from the symbol node updating during the last decoding iteration of the iterative decoding processing.

The method then continues by performing the symbol node updating. This symbol node updating involves computing the possible soft estimates of the symbols of the signal. The symbol node updating also performs updating of the bit edge messages using both the computed symbol metrics and the check ~~bit~~ edge messages passed by the last iteration of the iterative decoding processing."

6. On page 58, line 23 to page 59, line 6, please amend as follows:

"The method then continues by performing the check node updating. This check node updating involves updating the check ~~bit~~ edge messages using the bit edge messages passed from the symbol node updating during the last decoding iteration of the iterative decoding processing.

The method then continues by performing the symbol node updating. This symbol node updating involves computing the possible soft estimates of the symbols of the signal. The symbol node updating also performs updating of the bit edge messages using both the computed symbol metrics and the check ~~bit~~ edge messages passed by the last iteration of the iterative decoding processing.

Where this embodiment departs from the embodiment of the FIG. 28 is that the iterative decoding processing involves generating bit estimates during each decoding iteration. For example, this may be performed by doing hard limiting of the estimate of the symbol made from the possible soft symbol estimates. Then, once these hard bit estimates are provided, then the method determines whether the syndromes of the LDPC code are all equal to zero (within a certain degree of precision)."